NAG Library Routine Document

S10ACF

Note: before using this routine, please read the Users' Note for your implementation to check the interpretation of ***bold italicised*** terms and other implementation-dependent details.

1 Purpose

S10ACF returns the value of the hyperbolic cosine, $\cosh x$, via the function name.

2 Specification

```
FUNCTION S10ACF (X, IFAIL)
REAL (KIND=nag_wp) S10ACF
INTEGER            IFAIL
REAL (KIND=nag_wp) X
```

3 Description

S10ACF calculates an approximate value for the hyperbolic cosine, $\cosh x$.

For $|x| \leq E_1$, $\quad \cosh x = \frac{1}{2}(e^x + e^{-x})$.

For $|x| > E_1$, the routine fails owing to danger of setting overflow in calculating e^x. The result returned for such calls is $\cosh E_1$, i.e., it returns the result for the nearest valid argument. The value of machine-dependent constant E_1 may be given in the Users' Note for your implementation.

4 References

Abramowitz M and Stegun I A (1972) *Handbook of Mathematical Functions* (3rd Edition) Dover Publications

5 Parameters

1: X – REAL (KIND=nag_wp) *Input*

On entry: the argument x of the function.

2: IFAIL – INTEGER *Input/Output*

On entry: IFAIL must be set to 0, -1 or 1. If you are unfamiliar with this parameter you should refer to Section 3.3 in the Essential Introduction for details.

For environments where it might be inappropriate to halt program execution when an error is detected, the value -1 or 1 is recommended. If the output of error messages is undesirable, then the value 1 is recommended. Otherwise, if you are not familiar with this parameter, the recommended value is 0. **When the value -1 or 1 is used it is essential to test the value of IFAIL on exit.**

On exit: IFAIL $= 0$ unless the routine detects an error or a warning has been flagged (see Section 6).

6 Error Indicators and Warnings

If on entry IFAIL $= 0$ or -1, explanatory error messages are output on the current error message unit (as defined by X04AAF).

Errors or warnings detected by the routine:

IFAIL $= 1$

The routine has been called with an argument too large in absolute magnitude. There is a danger of overflow. The result returned is the value of $\cosh x$ at the nearest valid argument.

7 Accuracy

If δ and ϵ are the relative errors in the argument and result, respectively, then in principle

$$\epsilon \simeq x \tanh x \times \delta.$$

That is, the relative error in the argument, x, is amplified by a factor, at least $x \tanh x$. The equality should hold if δ is greater than the ***machine precision*** (δ is due to data errors etc.) but if δ is simply a result of round-off in the machine representation of x then it is possible that an extra figure may be lost in internal calculation round-off.

The behaviour of the error amplification factor is shown by the following graph:

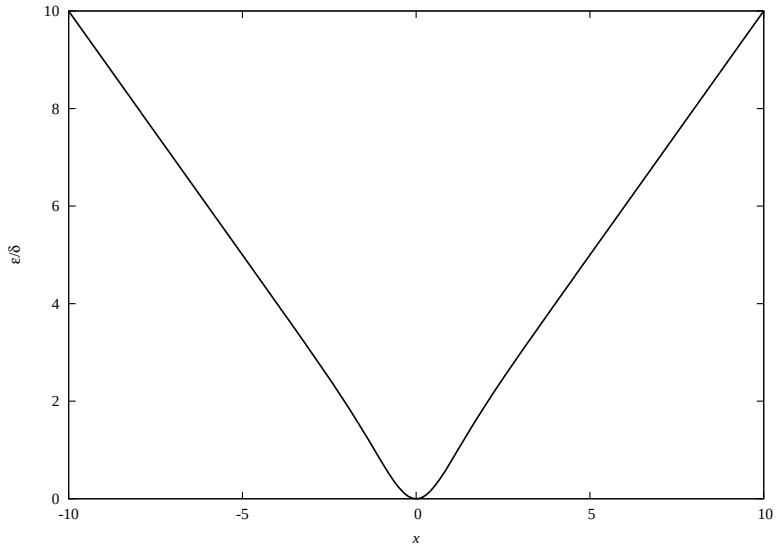

Figure 1

It should be noted that near $x = 0$ where this amplification factor tends to zero the accuracy will be limited eventually by the ***machine precision***. Also for $|x| \geq 2$

$$\epsilon \sim x\delta = \Delta$$

where Δ is the absolute error in the argument x.

8 Further Comments

None.

9 Example

This example reads values of the argument x from a file, evaluates the function at each value of x and prints the results.

9.1 Program Text

```
    Program s10acfe

!     S10ACF Example Program Text

!     Mark 24 Release. NAG Copyright 2012.

!     .. Use Statements ..
      Use nag_library, Only: nag_wp, s10acf
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: nin = 5, nout = 6
!     .. Local Scalars ..
      Real (Kind=nag_wp)               :: x, y
      Integer                          :: ifail, ioerr
!     .. Executable Statements ..
      Write (nout,*) 'S10ACF Example Program Results'

!     Skip heading in data file
      Read (nin,*)

      Write (nout,*)
      Write (nout,*) '     X           Y'
      Write (nout,*)

data: Do
        Read (nin,*,Iostat=ioerr) x

        If (ioerr<0) Then
          Exit data
        End If

        ifail = -1
        y = s10acf(x,ifail)

        If (ifail<0) Then
          Exit data
        End If

        Write (nout,99999) x, y
      End Do data

99999 Format (1X,1P,2E12.3)
    End Program s10acfe
```

9.2 Program Data

```
S10ACF Example Program Data
           -10.0
            -0.5
             0.0
             0.5
            25.0
```

9.3 Program Results

```
 S10ACF Example Program Results

      X           Y

   -1.000E+01   1.101E+04
   -5.000E-01   1.128E+00
    0.000E+00   1.000E+00
    5.000E-01   1.128E+00
    2.500E+01   3.600E+10
```